REMARKS

The claims are claims 1 to 15.

Claims 1, 3, 7, 8 and 10 have been amended to change "port" to "data port" and "ports" to "data ports" throughout. This change is made for consistency. Claim 8 has been further amended to include a ";" (semicolon) at the end of the first paragraph.

Claims 1 to 15 were rejected under 35 U.S.C. 102(e) as anticipated by Robertson et al U.S. Patent No. 6,496,740.

Claims 1 and 8 recite subject matter not anticipated by Robinson et al. Claim 1 recites "wherein at least one of said plurality of data ports consists of an active data port connected to said request queue controller capable of supplying a data transfer request to said request queue controller specifying a data source, a data destination and a data quantity to be transferred." Claim 8 similarly recites "wherein at least one of said plurality of data ports is an active data port; and generating a data transfer request at an active data port specifying a data source, a data destination and a data quantity to be transferred." Earlier recitations in these claims required the data ports to be connected to the hub for data transfer. Claims 1 and 8 each recite transfer requestor nodes for generating the data transfer requests. Thus claims 1 and 8 make a clear distinction between transfer requestor nodes and data ports. This application illustrates in Figure 4 transfer requestor nodes 116 connected to the chain transfer request bus nodes 117 and hence to request queue controller 101 and data ports 112, 112, 114 and 411. Data port 411 is the claimed active data port. Figure 4 illustrates a connection from data port 411 to special transfer request node 160 which is also connected to the chain of transfer request bus nodes 117 to transmit a data transfer request to request queue controller 101. Robinson et al fails to teach a data port connected to the hub for data transfer

and connected to the request queue controller for specifying data to be transferred by data source, data destination and data quantity.

The Examiner appears to be confusing data transfer from one data port through the hub to another data port with a node generating data transfer requests. Both this application and Robinson et al make clear that these are separate structures. This is summarized in the following tables. Table 1 lists the data ports recited in claims 1 and 8 and disclosed in Robinson et al.

Data Port		
Claim 1	Claim 8	Robertson et al
a plurality of data ports having an interior interface connected to the data transfer hub which is so configured as to be the same for each data port and an exterior interface configured for an external memory/device	transferring data from a source data port selected from a plurality of data ports corresponding to the data source specified in a data transfer request to a destination data port selected from said plurality of data ports	Figure 3: ports 350 to 355 Figure 4: external port interfaces 440 to 447 Figure 5: external port interface units 530 to 533

Table 1

Table 2 lists the transfer requestor nodes recited in claims 1 and 8 and disclosed in Robinson et al.

Transfer Requestor Nodes		
Claim 1	Claim 8	Robertson et al
at least one transfer requestor node connected to said request queue controller and capable of supplying a data transfer request to said request queue controller	generating via at least one transfer requestor node data transfer requests each specifying a data source, a data destination and a data quantity to be transferred	Figure 5: processor-cache (IMP Node) 570 to 572

Table 2

Table 3 lists the active data ports recited in claims 1 and 8.

Active Data Port		
Claim 1	Claim 8	Robertson et al
wherein at least one of said plurality of data ports consists of an active data port connected to said request queue controller capable of supplying a data transfer request to said request queue controller specifying a data source, a data destination and a data quantity to be transferred	wherein at least one of said plurality of data ports is an active data port; and generating a data transfer request at an active data port specifying a data source, a data destination and a data quantity to be transferred	

All of the citations to Robertson et al in the OFFICE ACTION teach either data ports or transfer requestor nodes. None of the citations to Robertson et al teach a single structure can perform both functions. In summary, Robinson et al fails to anticipate

that a single structure (called an active data port in claims 1 and 8) can both be a source and a destination for data and generate data transfer requests.

Robertson et al states at column 6, lines 21 to 40 (cited in the OFFICE ACTION):

> "Figure 3 illustrates the basic principal features of the TCHP. The TCHP is basically a data transfer controller which has at its front end portion, a request queue controller 300 receiving, prioritizing, and dispatching data in the form of transfer request packets. The request queue controller 300 connects within the hub unit 310 to the channel registers 320 which receive the data transfer request packets and processes them first by prioritizing them and assigning them to one of the N channels each of which represent a priority level. These channel registers interface with the source 330 and destination 340 pipelines which effectively are address calculation units for source (read) and destination (write) operations.
>
> "Outputs from these pipelines are broadcast to M Ports (six shown in Figure 3 as 350 through 355), which are clocked either at the main processor clock frequency or at a lower external device clock frequency. Read data from one port, e.g. port 350, having a destination write address of port 353 is returned to the hub destination control pipeline through the data router unit 360."

This portion of Robertson et al discloses: (a) request queue controller 300 receives, prioritizes and dispatches transfer request packets (column 6, lines 22 to 24; channel registers 320 receive the data transfer requests packets and assigns them to a channel (column 6, line 24 to 30); source pipeline 330 and destination pipeline 340 calculate addresses for source (read) and destination(write) (column 6, lines 30 to 33); these addresses are broadcast to ports 350 to 355 (column 6, lines 34 to 37); and data is read from a read port to data router unit 360, then written to a destination port (column 6, lines 37 to 41). This disclosure deals solely with data transfer and includes no teaching regarding data transfer request movement. This disclosure of Robertson et al

teaches the following connections to ports 350 to 355: source (read) addresses from source control pipeline 330; destination (write) addresses from destination control pipeline 340; read data from a read port to data router unit 360; and write data from data router unit 360 to a write port. None of these connections involve the data transfer request "specifying a data source, a data destination and a data quantity to be transferred" recited in claims 1 and 8. The Applicants submit that for a port 350 to 350 of Robertson et al to generate such a data transfer request requires a connection between that port and the "TRANSFER REQUEST PACKETS" input of request queue controller 300. Figure 3 of Robertson et al teaches no such connection.

The other portions of Robertson et al cited in the OFFICE ACTION likewise fail to disclose this subject matter. Figure 5 of Robertson et al illustrates ports 521 including port interface units 530, 531, 532 and 533. These are illustrated as connected to TCHP Hub 520, which includes a queue manager. Figure 5 of Robertson et al also illustrates processor-cache (IMP nodes) 570, 571 and 572 which are connected to transfer request feed mechanism 545. Figure 5 illustrates that transfer request feed mechanism 545 supplies requests to the queue manager part of TCHP hub 520. Robertson et al always discloses that requests come from processor-cache (IMP nodes) 570, 571 and 572 and never discloses that requests come from ports 521 or external port interface units 530, 531, 532 and 533. Thus Robertson et al fails to teach the active data port recited in claims 1 and 8. Robinson et al states at column 11, lines 9 to 28:

> "FIG. 5 shows from a higher level, the interconnection of the four main functional blocks, the first two of which, the TCHP hub 520 and the ports 521 (including all ports interface composed in FIG. 5 of 530, 531, 532, 533 and 560). The TCHP hub 520 and the ports 521 are an integral part of the TCHP. The other two units, the transfer request feed mechanism 545

and the data transfer bus DTB 555 are closely associated functional units, but not a part of the TCHP itself. FIG. 5 highlights the possible connection multiple data transfer bus DTB nodes and the possible connection of multiple transfer request nodes.

"Address Generation

"Address and word count are required outputs of each address unit and these outputs update the selected channel, given the size of the transfer to be performed. The complexity of address generation within the TCHP is increased by the need to accommodate the following transfer options which are of two major types:

"1. Normal Linear transfers, and

"2. two dimensional (2-D) transfers."

This portion of Robinson et al discloses ports 521 including external port interface units 530, 531, 532 and internal memory port master 560 illustrated in Figure 5. These are all connected to TCHP hub 520. Figure 5 also illustrates processor-cache (IMP nodes) 570, 571 and 572. These are all connected to data transfer bus 555 and transfer request feed mechanism 545. Figure 5 fails to illustrate any structure connected to both the TCHP hub 520 and the queue manager. Such a connection is required for the active data port recited in claims 1 and 8.

Column 12, lines 43 to 55 of Robinson et al teaches parts of the request queue manager in the form of a queue manager request bus master as illustrated in Figure 7. This structure corresponds to the request queue controller recited in claim 1 and the receiving, prioritizing and dispatching data transfer requests step recited in claim 8. This portion of Robinson et al includes no teaching regarding data ports and no teaching that a data port is connected to both the data transfer hub and the request queue manager as required by claims 1 and 8. The reference to Figure 8 of Robinson et al at column 12, lines 49 to 55 likewise fails to provide any teaching of a data port. Claims 1 and 8 are not anticipated by Robertson et al because Robertson et al fails to teach this limitation.

The OFFICE ACTION includes two recitations "wherein selected control line is implies active port" at page 4, line 7 and page 12. The OFFICE ACTION fails to point out where Robertson et al teaches this "control line." The Applicants respectfully submit that a careful study of Figures 3 and 5 of Robertson et al reveals no such "control line." In the absence of any indication within the OFFICE ACTION what disclosure of Robertson et al supports this "control line," this argument cannot be persuasive.

The RESPONSE TO AMENDMENT at paragraph 4b includes the statement:

> "In response to applicant's argument that Robertson includes no teaching regarding ports and no teaching that a port is connected to both the data transfer hub and the request queue manager. Examiner respectfully disagrees."

This misquotes the argument in the response filed January 19, 2005. The response filed January 19, 2005 states at page 10, lines 25 to 28:

> "This portion of Robinson et al includes no teaching regarding ports and no teaching that a port is connected to both the data transfer hub and the request queue manager as required by claims 1 and 8."

The "This portion" noted in this sentence is column 12, lines 43 to 55 and Figure 7 of Robinson et al, which is cited at the beginning of the paragraph at page 10, lines 20 to 22 of that response. As noted in Robertson et al at column 7, lines 34 to 36, Figure 7 of Robinson et al illustrates the queue manager. This queue manager corresponds to the request queue controller recited in claim 1 and the "receiving, prioritizing and dispatching" step recited in claim 8. Thus column 12, lines 43 to 55 and Figure 6 of Robinson et al fails to teach data ports.

Claims 2 and 9 recite subject matter not anticipated by Robertson et al. Claim 2 recites "said active data port capable of generating a data transfer request specifying said active data port as said data destination." Claim 9 recites "generating a data transfer request at an active data port specifying said active data port as said data destination." Robertson et al discloses ports 350 to 355 (Figure 3) and 530 to 533 (Figure 5) that can be data destinations and nodes 570 to 572 that can specify data transfer requests. However, Robertson et al fails to disclose the recited active data port which can both be a data destination and generate a data transfer request specifying itself as the data destination. Accordingly, claims 2 and 9 are allowable over Robertson et al.

Claims 4 and 11 recite subject matter not anticipated by Robertson et al. Claim 4 recites "said active data port capable of generating a data transfer request specifying said active data port as said data source." Claim 11 recites "generating a data transfer request at an active data port specifying said active data port as said data source." Robertson et al discloses ports 350 to 355 (Figure 3) and 530 to 533 (Figure 5) that can be data sources and nodes 570 to 572 that can specify data transfer requests. However, Robertson et al fails to disclose the recited active data port which can both be a data source and generate a data transfer request specifying itself as the data source. Accordingly, claims 4 and 11 are allowable over Robertson et al.

Claims 14 and 15 recite subject matter not anticipated by Robinson et al. Claim 14 recites "a special transfer request node connected to said upstream most node of said plurality of transfer request nodes and said active data port, said special transfer request node connecting said active data port to said request queue controller via said plurality of transfer request nodes." Claim 15 recites "transferring data transfer requests from said active data port to said request queue controller via a special transfer

request node connected to said upstream most transfer request node." This is illustrated in the application at Figure 4 which shows the connection of active data port 411 to special transfer request node 160 at the upstream most end of the chain of transfer request nodes 117. Robertson et al teaches something like the plurality of transfer request nodes in the transfer request feed mechanism 545 illustrated in Figure 5. Figure 5 of Robertson et al fails to show any connection of a data port to this transfer request feed mechanism 545. Thus, no teaching of Robinson et al anticipates this subject matter.

The OFFICE ACTION cites column 7, line 64 to column 8, line 33 and column 9, line 61 to column 10, line 18 as anticipating this subject matter. This is another example of the Examiner confusing data transfer with movement of data transfer requests. The cited portions of Robertson et al include repeated references to the transfer controller hub (TC hub). This is disclosed in Robertson et al as operating for data transfer. In contrast, claims 14 and 15 recite movement of data transfer requests. The cited portions of Robertson et al are not applicable to this different limitation in claims 14 and 15. Accordingly, claims 14 and 15 are not anticipated by Robertson et al.

The Applicants respectfully submit that all the present claims are allowable for the reasons set forth above. Therefore early reconsideration and advance to issue are respectfully requested.

If the Examiner has any questions or other correspondence regarding this application, Applicants request that the Examiner contact Applicants' attorney at the below listed telephone number and address to facilitate prosecution.

Texas Instruments Incorporated
P.O. Box 655474 M/S 3999
Dallas, Texas 75265
(972) 917-5290
Fax: (972) 917-4418

Respectfully submitted,

Robert D. Marshall, Jr.
Reg. No. 28,527